09/522510

cof ✓

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Youfeng Wu	Examiner: Wood, William
Patent No.: 7,100,155	Group Art Unit: 2193
Issue Date: August 29, 2006	Docket No: 884.258US1

Title: SOFTWARE SET-VALUE PROFILING AND CODE REUSE

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450
ATTN: CERTIFICATE OF CORRECTION BRANCH

We are transmitting herewith the attached:

X Authorization to charge Deposit Account No. 19-0743 in the amount of $100.00 to cover the Certificate of Correction fee.
X Request for Certificate of Correction.
X Certificate of Correction Form - PTO-1050 (1 page)
X A return postcard.

Please charge any additional fees or credit overpayment to Deposit Account No. 19-0743.

SCHWEGMAN, LUNDBERG & WOESSNER P.A.
Customer No: 21186

By:
Name: Charles E. Steffey
Reg. No.25,179
CES:CMG:sj

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail, in an envelope addressed to: Commissioner of Patents and Trademarks, P.O. Box 1450, Alexandria VA 22313-1450, on this 19 day of May 2008.

Richard Beck
Name

Richard Beck
Signature

Patent 7,100,155 PATENT

IN UNITED STATES PATENT AND TRADEMARK OFFICE

Patent No.: 7,100,155 Docket No: 884.258US1
Issue Date: August 29, 2006 Patentee: Youfeng Wu
Customer No.: 21186 Confirmation No.: 5136
Title SOFTWARE SET-VALUE PROFILING AND CODE REUSE

REQUEST FOR CERTIFICATE OF CORRECTION

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450
ATTN: CERTIFICATE OF CORRECTION BRANCH

It is requested that a Certificate of Correction be issued correcting printing errors appearing in the above-identified United States patent. A copy of the text of the Certificate in the suggested form are enclosed.

Pursuant to 1.20(a), please charge Deposit Account No. 19-0743 in the amound of $100.00.

Issuance of the Certificate of Correction would neither expand nor contract the scope of the claims as properly allowed, and re-examination is not required.

The Examiner is authorized to charge any additional fees or credit overpayment to Deposit Account No.19-0743.

Respectfully Submitted
YOUFENG WU
By his Representatives,

SCHWEGMAN, LUNDBERG & WOESSNER P.A
P.O. Box 2938
Minneapolis, MN 55402
(612) 373-6900

Date : May 16, 2008

By: Charles Steffey
Charles E. Steffey
Reg. No: 25,179
CES:sj

CERTIFICATE UNDER 37 CFR § 1.8: The undersigned hereby certifies that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail, in an envelope addressed to: Commissioner for Patents, P.O. Box 1450 Alexandria, VA 22313-1450, on this 14 day of May 2008

Richard Beek
Name

Richard Beck
Signature

PTO/SB/44 (02-01)
PTO/SB/44 (02-01)
Approved for use through 01/31/2004. OMB 0651-0033
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.
(Also Form PTO-1050)

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO : 7,100,155 Page (1) of 1

DATED : August 29, 2006

INVENTOR(S) : Wu

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

On the face page, in field (56), under “Other Publications”, in column 2, line 13, delete “Internatinal” and insert - - International - -, therefor.

In column 10, line 31, in Claim 2, after “claim 1” insert - - , - -.

In column 12, line 3, in Claim 17, after “claim 16” insert - - , - -.

In column 12, line 53, in Claim 25, delete “set values” and insert - - set-values - -, therefor.

MAILING ADDRESS OF SENDER:

SCHWEGMAN, LUNDBERG & WOESSNER P.A.
P.O. BOX 2938
Minneapolis, MN 55402

Atty Docket No: 884.258US1

PATENT NO. ______7,100,155______

No. of additional copies

SCHWEGMAN LUNDBERG WOESSNER KLUTH

Issued Patent Proofing Form **SLWK: 884.258US1**
Note: P = PTO Error **S = SLWK Error** **Proofread By: Sarika (09/07/2006)**

US Serial No.: 09/522,510 **US Patent No.: US 7,100,155 B1** **Issue Dt.: Aug. 29, 2006**
Title: SOFTWARE SET-VALUE PROFILING AND CODE REUSE
PR Instructions: Face Page, Claims and Abstract

Sr. No.	P/S	Original		Issued Patent		Description of Error
		Page	Line	Column	Line	
1	P	Page 2 of 3 List of references cited by examiner (02/13/2003)	Entry 1 Line 1 (Non-Patent Documents)	First Page Col. 2 (Other Publications)	13	Delete "Internatinal" and insert - - International - -, therefor.
2	P	Page 2 Amendment - After Non-Final Rejection (In the Claims) (07/13/2005)	Claim 6 Line 1	10	31 (Approx.)	In Claim 2, after "claim 1" insert - - , - -.
3	S	Page 5 Amendment - After Non-Final Rejection (In the Claims) (07/13/2005)	Claim 19 Line 1	12	3	In Claim 17, after "claim 16" insert - - , - -.
4	S	Page 7 Amendment - After Non-Final Rejection (In the Claims) (07/13/2005)	Claim 24 Line 2	12	53	In Claim 25, delete "set values" and insert - - set-values - -, therefor.

US007100155B1

(12) **United States Patent**
Wu

(10) **Patent No.: US 7,100,155 B1**
(45) **Date of Patent: Aug. 29, 2006**

(54) **SOFTWARE SET-VALUE PROFILING AND CODE REUSE**

(75) Inventor: **Youfeng Wu**, Palo Alto, CA (US)

(73) Assignee: **Intel Corporation**, Santa Clara, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/522,510**

(22) Filed: **Mar. 10, 2000**

(51) **Int. Cl.**
G06F 9/45 (2006.01)
G06F 9/44 (2006.01)

(52) **U.S. Cl.** **717/158**; 717/130

(58) **Field of Classification Search** 717/127, 717/130–132, 140, 151–158, 145, 159
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,355,487 A	*	10/1994	Keller et al.	717/127
5,845,101 A		12/1998	Johnson et al.	395/383
5,845,103 A		12/1998	Sodani et al.	395/392
5,909,578 A	*	6/1999	Buzbee	717/130
5,933,628 A	*	8/1999	Chang	712/233
5,960,198 A	*	9/1999	Roediger et al.	717/130
6,253,373 B1	*	6/2001	Peri	717/150
6,389,468 B1	*	5/2002	Muller et al.	709/226
6,427,234 B1	*	7/2002	Chambers et al.	717/140

OTHER PUBLICATIONS

Wu et al., "Better Exploration of Region-Level Value Locality with Integrated Computation Reuse and Value Prediction", IEEE, 2001, 98-108.*

Watterson et al., "Goal-Directed Value Profiling", Springer-Verlag Berlin Heidelberg, 2001, pp. 319-333.*

Sastry et al., "Characterizing Coarse-Grained Reuse of Computation", University of Wisconsin, Madison.*

Sodani et al., "Dynamic Instruction Reuse", ACM, 1997.*

Hall, "Call Path Profiling", ACM, 1992, pp. 296-306.*

Calder et al., "Value Profiling and Optimization", Journal of Instruction-Level Parallelism 1, 1999.*

Connors et al., "Compiler-Directed Dynamic Computation Reuse: Rationale and Initial Results", IEEE, 1999, pp. 158-169. *Retransmitted for page numbered copy.*

"Dictionary of Computing", Oxford University Press, Market House Books, 1996.*

Molina et al., "Dynamic Removal of Redundant Computations", Internatinal Conference on Supercomputing, 1999.*

Lilja et al., "Exploiting Basic Block Value Locality with Block Reuse", 1998.*

Gonzalez et al., "Trace-Level Reuse", 1999.*

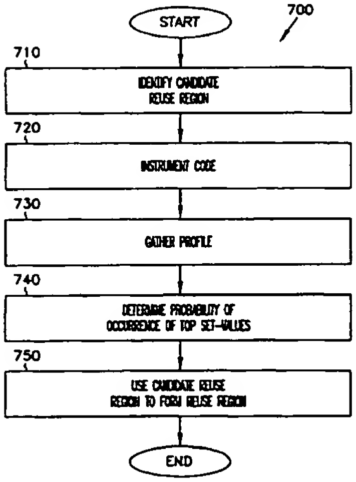
(Continued)

Primary Examiner—Kakali Chaki
Assistant Examiner—William H. Wood
(74) *Attorney, Agent, or Firm*—Schwegman, Lundberg, Woessner & Kluth, P.A.

(57) **ABSTRACT**

An apparatus and method for profiling candidate reuse regions and candidate load instructions aids in the selection of computation reuse regions and computation reuse instructions with good reuse qualities. Registers holding input values for candidate reuse regions are sampled periodically when the candidate reuse region is encountered. The register contents are combined into set-values. When a relatively small number of set-values account for a large percentage of occurrences, the candidate reuse region may be a good computation reuse region. Load instructions are profiled for the location accessed and the value loaded. The location and value are combined into location-values. The relative occurrence frequency of location-values can be used to evaluate load instructions as candidate instructions for reuse.

33 Claims, 6 Drawing Sheets


START
700
710
IDENTIFY CANDIDATE REUSE REGION
720
INSTRUMENT CODE
730
GATHER PROFILE
740
DETERMINE PROBABILITY OF OCCURRENCE OF TOP SET-VALUES
750
USE CANDIDATE REUSE REGION TO FORM REUSE REGION
END

number of criteria. One such candidate reuse region is shown as candidate reuse region **100** in FIG. **1**.

In action **720**, the software program code is instrumented for profiling. Instrumenting for profiling includes inserting instructions in the program that profile top set-values and top location-values. In some embodiments, every time a candidate reuse region is encountered, the instrumented code profiles a set-value for the candidate reuse region. In other embodiments, a sampling filter is employed, such as filter **504** (FIG. **5**A), and only one of every "S" set-values is profiled.

In action **730**, the instrumented code is executed and the profile data is gathered. As a result, profiling data structures, such as profiling data structure **400** (FIG. **4**), and profiling data structure **650** (FIG. **6**D) are generated. In action **740**, the probability of occurrence of a top set-value is determined as the ratio of the number of times the top set-value was collected to the total number of times set-values were sampled. When a small number of top set-values represent a large percentage of the execution of the candidate reuse region, then the candidate reuse region will likely make for a good computation reuse region.

In action **750**, the candidate reuse region is used to form a computation reuse region if appropriate criteria are met. One such criteria is when the probability of occurrence of a small number of top set-values exceeds a threshold. A candidate reuse region can be used by itself or can be combined with other candidate reuse regions to form a computation reuse region.

FIG. **8** shows a processing system. Processing system **800** includes processor **820** and memory **830**. In some embodiments, processor **820** is a processor capable .of executing instrumented software for profiling top set-values and top location-values. Processor **820** can also be a processor capable of selecting good computation reuse regions from candidate reuse regions. Processing system **800** can be a personal computer (PC), mainframe, handheld device, portable computer, set-top box, or any other system that includes software. In some embodiments, the processor includes one or more predicate registers **840**.

In some embodiments, processor **820** includes cache memory, a memory controller, or a combination of the two. In these embodiments, processor **820** may access a profile indicator data structure without accessing memory **830**. In other embodiments, profile indicators are maintained within memory **830**, and processor **820** accesses memory **830** when updating profile indicators regardless of whether processor **820** includes cache memory or memory controllers.

Memory **830** can be a random access memory (RAM), read only memory (ROM), flash memory, hard disk, floppy disk, CDROM, or any other type of machine medium readable by processor **820**. Memory **830** can store instructions for performing the execution of the various method embodiments of the present invention.

CONCLUSION

A software profiling mechanism that gathers and profiles top set-values and top location-values has been described. Software to be profiled is instrumented with instructions that sample set-values at the occurrence of candidate reuse regions and sample location-values at the occurrence of candidate load instructions. Set-values and location-values can be generated as concatenated values, or can be combined using mechanisms such as exclusive-or operators. When a small number of top set-values account for a large percentage of occurrences, the candidate reuse region may make a good computation reuse region. Likewise, when a small number of top location-values account for a large percentage of occurrences of candidate load instructions, the candidate load instruction may make a good candidate for inclusion in a computation reuse region.

It is to be understood that the above description is intended to be illustrative, and not restrictive. Many other embodiments will be apparent to those of skill in the art upon reading and understanding the above description. The scope of the invention should, therefore, be determined with reference to the appended claims, along with the full scope of equivalents to which such claims are entitled.

What is claimed is:

1. A computer-implemented method comprising:

identifying a candidate reuse region of a software program;

determining an input set for the candidate reuse region, wherein the input set includes a plurality of input registers for storing input values of the candidate reuse region;

instrumenting the software program to, when executed, sample set-values for the input set, wherein each set-value includes an input register value for each of the plurality of input registers;

executing the instrumented software;

tracking, during the execution, a number of times a set-value is encountered; and

selecting, based on the tracking, the candidate reuse region as a computation reuse region.

2. The computer-implemented method of claim **1** wherein the input-set comprises a plurality of input registers, and each set-value comprises an input register value for each of the plurality of input registers, and wherein the instrumenting of the software program includes,

.inserting combine instructions into the software program, the combine instructions which, when executed, will combine each of the input register values into a single value; and

inserting index instructions into the software program, the index instructions which, when executed, will index into a data structure of profile indicators using the single value.

3. The computer implemented method of claim **1**, wherein the instrumenting of the software program includes inserting profile instructions to profile the top N occurring set-values, where N is based on a function of an expected number of reuse instances.

4. A machine readable medium including instructions, which when executed by a machine, cause the machine to perform operations according to the computer implemented method of claim **1**.

5. The machine readable medium of claim **4**, wherein, during the execution, the sampling is performed every S occurrences of the set-values, and wherein S is an integer greater than **1**.

6. The machine readable medium of claim **4** further including instructions, which when executed by a machine, cause the machine to, for each set-value, combine each of the input register values into a single value.

7. The computer implemented method of claim **1**, wherein during the execution, the sampling is performed every S occurrences of the set-values, and wherein S is an integer greater than **1**.

8. The computer implemented method of claim 1 further comprising, for each set-value, combining each of the input register values into a single value.

9. The computer-implemented method of claim 8, wherein the combining of each of the input register values into a single value includes:

folding each of the input register values to create folded values; and

concatenating the folded values.

10. A computer-implemented method comprising:

determining whether a software program region is a computation reuse region, wherein the determining includes,

periodically sampling a set of registers to obtain register values, wherein the register values are input values of the software program region;

combining the register values into a single set-value;

determining an occurrence frequency of the single set-value; and

storing the occurrence frequency and the single set-value in a data structure;

basing the determination of whether the software program region is the computation reuse region on the occurrence frequency.

11. The computer-implemented method of claim 10, wherein the periodically sampling of the set of registers includes sampling ones of the set of registers to obtain a set-value every S occurrences of the software program region, wherein S is a sampling period, wherein S is greater than 1, and wherein S is chosen so that a statistically valid number of registers are sampled.

12. The computer-implemented method of claim 11 further comprising:

identifying a group of control equivalent candidate region entries and candidate load instructions;

inserting predicate instructions prior to ones of the group, wherein the predicate instructions set a predicate register every S occurrences; and

inserting profiling instructions at each of the control equivalent candidate region entries and candidate load instructions, wherein the profiling instructions are predicated on the predicate register.

13. The computer-implemented method of claim 11, wherein the storing includes,

accessing a record in the data structure as a function of the set-value; and

incrementing a profile indicator associated with the record.

14. The computer-implemented method of claim 11, wherein the periodically sampling of the set of registers further includes sampling, at the beginning of a candidate reuse region, set-values in ones of the set of registers, the plurality of registers being input registers to the candidate reuse region.

15. A computer-implemented method comprising:

identifying a candidate load instruction in a software program;

instrumenting the software program to, when executed, sample a location-value every S occurrences of the candidate load instruction, wherein S is an integer greater than 1;

storing an occurrence frequency of the location-value into a data structure; and

executing the software program.

16. The computer-implemented method of claim 15, wherein the instrumenting of the software program includes,

inserting count instructions in the software program to count a number of times the location-value is sampled; and

inserting track instructions in the software program to keep track of top location-values.

17. The computer-implemented method of claim 16□ wherein the candidate region includes a plurality of candidate load instructions, each of the plurality of load instructions being predicated on a common predicate register.

18. The computer-implemented method of claim 16, wherein the inserting of the track instructions to keep track of top location-values includes inserting sampling instructions configured to profile the top N occurrences of location-values, where N is an integer.

19. The computer-implemented method of claim 15 further comprising:

identifying a group of control equivalent candidate region entries and candidate load instructions in the software program;

inserting predicate instructions in the software program prior to ones of the group, wherein the predicate instructions set a predicate register every S occurrences; and

inserting profiling instructions in the software program at each of the control equivalent candidate region entries and candidate load instructions, wherein the profiling instructions are predicated on the predicated register.

20. A machine readable medium including instructions, which when executed by a machine, cause the machine to perform operations according to the computer implemented method of claim 15.

21. The machine readable medium of claim 20, wherein the instrumenting of the software includes inserting count instructions in the software to count a number of times the location-value is encountered.

22. The machine-readable medium of claim 20, wherein the instrumenting of the software includes inserting track instructions in the software program to keep track of top location-values.

23. The computer-implemented method of claim 15, wherein S is chosen so that a statistically valid number of location-values are sampled.

24. A computer-implemented method comprising:

selecting candidate reuse regions within a software program; and

selecting reuse regions from the candidate reuse regions, the selecting of the reuse regions including,

periodically sampling set-values for ones of the candidate reuse regions to produce a probability of occurrence of top set-values, wherein each of the set-values includes values of input registers for one of the candidate reuse regions; and

basing the selection of the reuse regions on the probability of occurrence of the top set-values.

25. The computer-implemented method of claim 24, wherein sampling the set values includes, representing each set-value as a single value; and accessing a data structure as a function of the single value to modify a profile indicator.

26. The computer-implemented method of claim 25, wherein the data structure is at least as large as a number of expected reuse instances.

27. The computer-implemented method of claim 24, wherein selecting the reuse regions further includes marking as reuse regions those candidate reuse regions having a finite number of set-values that have a probability of occurrence greater than a threshold.

28. A machine readable medium including instructions, which when executed by a machine, cause the machine to perform operations according to the computer implemented method of claim 24.